AFRL-VA-WP-TM-2003-3019

4-POWERS AIR REFUELING TECHNOLOGIES LONG TERM TECHNOLOGY PROJECT

System Specifications/Requirements Necessary for Interfacing with D-Six Simulation Environment

Glenn Williams

Control Simulation and Assessment Branch (AFRL/VACD)
Control Sciences Division
Air Vehicles Directorate
Air Force Research Laboratory, Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7542

MARCH 2003

Final Report for 01 June 2002 – 01 October 2002

AIR VEHICLES DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
March 2003	Final	06/01/2002 – 10/01/2002

4. TITLE AND SUBTITLE

4-POWERS AIR REFUELING TECHNOLOGIES LONG TERM TECHNOLOGY PROJECT

System Specifications/Requirements Necessary for Interfacing with D-Six Simulation Environment

5a. CONTRACT NUMBER IN-HOUSE

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER N/A

6. AUTHOR(S)

Glenn Williams

5d. PROJECT NUMBER N/A

5e. TASK NUMBER N/A

5f. WORK UNIT NUMBER N/A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Control Simulation and Assessment Branch (AFRL/VACD)
Control Sciences Division
Air Vehicles Directorate
Air Force Research Laboratory, Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7542

8. PERFORMING ORGANIZATION REPORT NUMBER

AFRL-VA-WP-TM-2003-3019

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Vehicles Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7542

10. SPONSORING/MONITORING AGENCY ACRONYM(S)

AFRL/VACD

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)

AFRL-VA-WP-TM-2003-3019

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT *(Maximum 200 Words)*

This white paper discusses a method of interfacing the D-Six Simulation Environment running under MS Windows 2000 on a 1.7GHz Pentium 4 PC with existing simulator cockpits that are available to ART LTTP members. For some insight into the rational for doing this, refer to the ART LTTP white paper titled "Universal Auto Air-to-Air Refueling Simulation."

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor) / 19b. TELEPHONE NUMBER *(Include Area Code)*
Unclassified	Unclassified	Unclassified	SAR	8	Glenn Williams / (937) 904-6546

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

White paper by AFRL/VAC for ART LTTP topic:

System Specifications/Requirements Necessary for Interfacing with D-Six Simulation Environment

This white paper discusses a method of interfacing the D-Six Simulation Environment running under MS Windows 2000 on a 1.7GHz Pentium 4 PC with existing simulator cockpits that are available to ART LTTP members. For some insight into the rational for doing this, refer to the ART LTTP white paper titled "Universal Auto Air-to-Air Refueling Simulation".

This paper discusses the interface between the computer the D-Six Simulation Environment is running on (D-6) and a high fidelity simulator cockpit. The methods used and lessons leaned on this project could be adapted by other ART LTTP members if a joint project were initiated by the working group and if it was decided to host the joint aerial refueling (AR) project on a D-6 system. The simulator cockpit that D-6 was interfaced to, provides a working example of the relative ease with which the interface can be accomplished. The cockpit chosen was conveniently very modularized and required no hardware or software changes to ether of it's two subsystems (input/output (I/O) and graphics generation). Before this project began, this modular cockpit system was interconnected via a Shared Common RAM Network (SCRAMNet) to computer(s) that were normally used to simulate the aircraft flight and missions. These computers were removed from the SCRAMNet loop and replaced with D-6. All that was required to interface D-6 to the simulated cockpit was to install a SCRAMNet PCI card into D-6 and to write the software modules for D-6 that would allow it to communicate with the cockpit I/O subsystem and the graphics generation subsystem. The primary software modules required were an MS Windows driver for the SCRAMNet PCI card and a module that handled the data transfer between the Windows OS and D-6. It was decided to have the driver written by an in-house contractor because the commercial driver was costly and we could foresee needing many copies of it in the future. The data transfer module was also written in-house and is an integral part the refueling control process. D-6, the cockpit I/O, and the graphics generator are all connected to the same SCRAMNet optical ring and each system has an exclusive memory partition in SCRAMNet that only it is allowed to write to, but that all computers on the ring are allowed to read. With the communication setup, all that D-6 had to do was read the cockpit I/O data from SCRAMNet (it was written to SCRAMNet by the cockpit I/O subsystem once per simulation frame) and to write the data to SCRAMNet required by the cockpit's graphic generator subsystem (heading, pitch, roll, x, y, z, etc. of each aircraft and the boom) once per simulation frame.

The specifications of the interface between D-6 and an external high resolution wide angle graphics display system and a simulated aircraft cockpit will be divided into three parts. The first part will outline the hardware implementation, and will explain what hardware was used and how it was configured. The second part will outline the communications implementation, and will explain what software modules are required in order to transfer the necessary data between the three subsystems. The third part discusses some issues relating to sharing D-6 software modules by ART LTTP members or other third party sources.

An outline of the major steps used to implement the hardware interface between the D-6 system and the existing graphics generator (GG) and cockpit I/O (CIO) subsystems follows :

- Install a PCI SCRAMNet (Shared Common RAM Network) card into the D-6 system. The GG and CIO each already had a SCRAMNet card and no change was required.
- Connect to the SCRAMNet cards in the D-6, the GG, and the CIO into the same optical ring. This will function as the communications link between these systems. Note that Ethernet may also work as the communications link, but has not been implemented and latency issues will have to be checked.

ASC- 03-0654

- D-6 requires a SCRAMNet driver for the MS Windows 2000 to operate the PCI SCRAMNet card the was installed in the machine. The GG and the CIO already had the require drivers and no change was required.

An outline of the major steps used to implement the communications interface between the D-6 system, the existing GG and CIO subsystems follows :

- D-6 requires a module that performs the necessary calls to the respective OS in order to read and write data via the SCRAMNet I/O card. D-6 has wizards that can assist in creating this module for D-6. The wizard will create a generic module of the type required and all you have to do is modify it's code to accomplish your specific tasks. The GG and CIO already had this module and no change was required.
- D-6 acquires the following cockpit data by reading data from SCRAMNet that was written to SCRAMNet by the CIO each frame of the simulation.
    - Stick pitch and roll
    - Throttle
    - Any other data items required from the cockpit
- The GG acquires the following model/flight data by reading data from SCRAMNet that was calculated and written to SCRAMNet by D-6 each frame of the simulation.
    - Heading, pitch, roll, X, Y, and Z of the tanker
    - Heading, pitch, and length of the boom
    - Heading, pitch, roll, X, Y, and Z of the four receivers
    - Data items to be displayed on a HUD
- D-6 and the CIO are synchronized on the D-Six frame number counter which is written to SCRAMNet by D-6 and read from SCRAMNet by the CIO.

Interfacing D-Six projects with software components from third party sources.

- The easiest and most flexible method is to share source code (C++ preferred) that has been developed for and debugged on a D-Six system.
- If the source code is not sharable, the next preferred method is to share compiled object code in the form of an object module, a linkable library or a dynamically linked library (DLL). The program interface must be agreed upon by all users and the object module must have been compiled with MS Windows 2000 as the target system. This code should be tested and debugged on a D-Six project that is common to all users prior to its distribution.
- The least desired method, but sometimes necessary in order to further protect the code; is to execute the code on a discrete, protected, and access controlled computer. The computers are then connected together via a data link (SCRAMNet, Ethernet, a link using the HLA protocol, etc.), and only the data put on the data link is visible to other users. The content and format of the data on the data link must be agreed upon by all users.
- The first two methods will result in the fastest system development because shared software can be transmitted via e-mail, ftp, or physical mail and each user can run the simulation on their time schedule. Whereas the third method requires operational personnel from the respective parties to meet at a common point or establish a remote link in order to run the complete simulation. This is much less convenient because it requires more personnel and they must be simultaneously available.